

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 11-219284  
 (43)Date of publication of application : 10.08.1999

BEST AVAILABLE COPY

51)Int.CI.

G06F 9/06  
G06F 13/00

21)Application number : 10-019402

(71)Applicant : SANYO ELECTRIC CO LTD

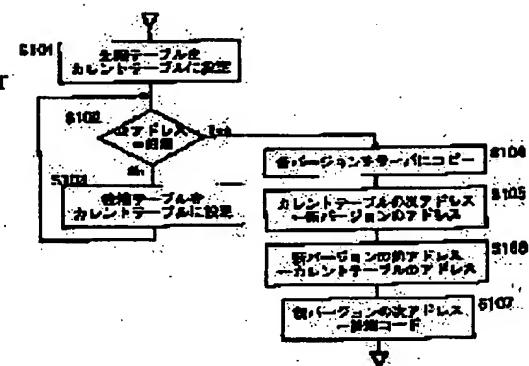
22)Date of filing : 30.01.1998

(72)Inventor : TANAKA YOSHIAKI

**54) VERSION UPDATING METHOD****57)Abstract:**

**PROBLEM TO BE SOLVED:** To manage plural version updating data coexistent in a server.

**SOLUTION:** In a client/server type LAN system, version updating data of a software introduced by a client are held in a server over a plural versions and the version updating data are connected in the order of the versions and managed. Also, the client refers to the version updating data held in the server and the version is successively updated to versions after the version of the software of the present client.

**LEGAL STATUS**

Date of request for examination]

Date of sending the examiner's decision of rejection]

Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

Date of final disposal for application]

Patent number]

Date of registration]

Number of appeal against examiner's decision of rejection]

Date of requesting appeal against examiner's decision of rejection]

Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

**THIS PAGE BLANK (USPTO)**

文書上

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平11-219284

(43)公開日 平成11年(1999)8月10日

(51)Int.Cl.  
G 06 F 9/06  
19/00

識別記号  
410  
351

F I  
G 06 F 9/06  
13/00

410 Q  
351 H

審査請求 未請求 請求項の数7 O.L. (全7頁)

(21)出願番号 特願平10-19402

(22)出願日 平成10年(1998)1月30日

(71)出願人 000001889

三洋電機株式会社

大阪府守口市京阪本通2丁目5番5号

(72)発明者 田中 義昭

大阪府守口市京阪本通2丁目5番5号 三  
洋電機株式会社内

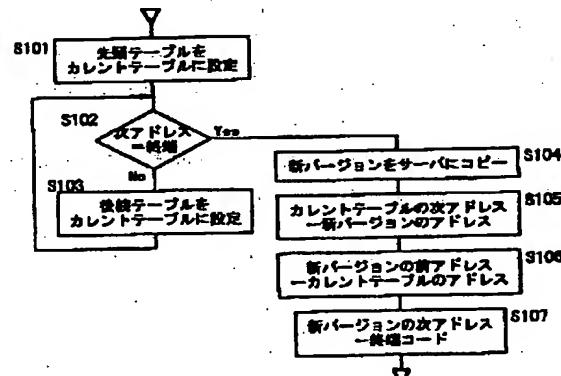
(74)代理人 弁理士 安富 耕二 (外1名)

(54)【発明の名称】 バージョンアップ方法

(57)【要約】

【課題】 サーバは、すべてのクライアントがバージョンアップを終了するまで、現在保持するソフトウェアを次のバージョンに更新することができない。

【解決手段】 クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理すると共に、クライアントはサーバに保持されたバージョンアップデータを参照して、自クライアントのソフトウェアのバージョン以降のバージョンに順次バージョンアップする構成である。



## 【特許請求の範囲】

【請求項1】 クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理することを特徴とするバージョンアップ方法。

【請求項2】 クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理すると共に、クライアントはサーバに保持されたバージョンアップデータを参照して、自クライアントのソフトウェアのバージョン以降のバージョンに順次バージョンアップすることを特徴とするバージョンアップ方法。

【請求項3】 クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理すると共に、クライアントが導入したソフトウェアのバージョン情報をサーバに保持するようになり、サーバはそのバージョン情報を参照してクライアントがバージョンアップを終了したバージョンアップデータを削除することを特徴とするバージョンアップ方法。

【請求項4】 クライアントは、サーバが保持するクライアントが導入したソフトウェアのバージョン情報を、バージョンアップ後の自クライアントのソフトウェアのバージョンに変更することを特徴とする請求項2または3に記載のバージョンアップ方法。

【請求項5】 サーバはバージョンアップデータを管理する際に、バージョンアップデータのバージョン情報と、バージョンアップデータの実体と、バージョン順のリンク情報をテーブルに記憶して管理することを特徴とする1乃至4のいずれかに記載のバージョンアップ方法。

【請求項6】 クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理すると共に、クライアントが導入したソフトウェアのバージョン情報をサーバに保持するようになり、クライアントはバージョンアップを行うとき、

- a. サーバに保持されたバージョンアップデータを参照して最も古いバージョンのバージョンアップデータを対象とし、
- b. 対象のバージョンアップデータのバージョンと自クライアントソフトのバージョンとを比較し、

c. 自クライアントソフトのバージョンが古い場合に、対象のバージョンアップデータをサーバから取り込んでバージョンアップを行い、

d. サーバが保持するバージョン情報を、前ステップで導入したバージョンアップデータのバージョンに変更し、

e. 後続するバージョンのバージョンアップデータを新たな対象とし、

f. ステップbに復帰して処理を繰り返す、ステップを有することを特徴とするバージョンアップ方法。

【請求項7】 クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理すると共に、クライアントが導入したソフトウェアのバージョン情報をサーバに保持するようになり、サーバはバージョンアップデータの削除を行うとき、

a. 最も古いバージョンのバージョンアップデータを対象とし、

b. 対象のバージョンアップデータのバージョンと、サーバが保持するクライアントが導入したソフトウェアのバージョン情報を比較し、

c. すべてのクライアントがそのバージョンに更新済みのとき、対象のバージョンアップデータを削除し、

d. 後続するバージョンのバージョンアップデータを新たな対象とし、

e. ステップbに復帰して処理を繰り返す、ステップを有することを特徴とするバージョンアップ方法。

## 【発明の詳細な説明】

## 【0001】

【発明の属する技術分野】 本発明は、クライアント/サーバ方式のLANシステム等において、クライアントのソフトウェアのバージョンアップデータをサーバ上で一元的に管理して、クライアントソフトウェアのバージョンアップを行う方法に関する。

## 【0002】

【従来の技術】 クライアント/サーバ方式のLANシステム等でクライアントのソフトウェアのバージョンアップを行う場合、個々のクライアントが、バージョンアップデータが記録されたCD-ROM等の記憶媒体からバージョンアップデータを読み取って、バージョンアップを行っていた。しかし、この方法ではそれぞれのクライアントで人手を要するので、コストがかかる問題があった。そこで、例えば、特開平9-292980号公報に記載のファイル配布システムのように、サーバの記憶装置にクライアントソフトの最新のバージョンアップデータを保持し、クライアントがそのデータを自動的にダウンロードして、バージョンアップを行う方法が行われている。

## 【0003】

【発明が解決しようとする課題】このようなシステムでは、クライアントのソフトウェアのバージョンアップが個々のクライアントの責任において行われるので、サーバに保持されたソフトウェアが最新のバージョンに更新されても、必ずしもすべてのクライアントが一時に一斉にバージョンアップを行って、そのソフトウェアを最新のバージョンに保っている保証はない。従って、サーバは、すべてのクライアントがバージョンアップを終了するまで、現在保持するソフトウェアを次のバージョンに更新することはできない。

【0004】また、それを防止するために、サーバが各クライアントを監視してクライアントソフトのバージョンアップを実施する方法もあるが、この方法は前記公報の従来技術にも記載されている通り、サーバの負荷が大きいという問題がある。従って、頻繁にバージョンアップを行う必要があるシステムでは、最新バージョンのみをサーバに保持する従来の方法ではうまく機能しない問題がある。

【0005】ところで、ソフトウェアのバージョンアップを行う場合、そのソフトウェア全体をそっくり新しいものに入れ替える方法もあるが、他の方法として、そのソフトウェアを構成する一部のプログラムファイルやデータファイルだけを更新する、部分的なバージョンアップも多用される。複数のプログラムファイルによって構成されたソフトウェアのうち、ある機能を受け持つプログラムだけを更新したり、ソフトウェア内部で参照するテーブルのデータだけを更新するのがそれである。こうすることでバージョンアップにかかる時間も費用も節約できるものである。これら更新部分は、ソフトウェア全体から見ると新バージョンと旧バージョンとの差に相当するので、差分ファイルや差分データとも呼んでいる。

【0006】例えば、病院の会計処理を行うソフトウェアでは、患者が受けた行為を入力してその診療報酬額や患者への請求額を算定する。その際、薬価の合計を計算するために処方薬の単価を記憶した薬価テーブルを参照する。また、ある種の診療行為については組み合わせる条件によって報酬額が変わってくるので、それを判定するプログラムが実行される。そして、これら薬価や組み合わせの条件は法令の改正に伴い変更され得るので、その時は、薬価テーブルを記憶したデータファイルだけ、あるいは、該当の判定プログラムが入ったプログラムファイルだけを、バージョンアップデータとしてクライアントに供給することが行われる。

【0007】しかし、クライアントにおいて、現行バージョンのソフトウェアで処理すべきデータが残っている場合は、供給された新バージョンの差分ファイルをインストールすることを控え、現行バージョンをそのまましばらく使い続けることもある。また、何らかの理由によってバージョンアップを行わない、あるいは行えないク

ライアントもある。そして、その間にもさらに新しいバージョンが出ることもあるが、この場合、2世代前のバージョンを飛ばして最新のバージョンだけをインストールすると不都合が生じる可能性がある。なぜなら、2世代前のバージョンアップの内容が最新バージョンの差分ファイルに反映されていない可能性があるからである。これが差分ファイルの意味であり、歴代の差分ファイルを順番にインストールすること、即ちすべてのバージョンでバージョンアップ作業をすることが必要である。そして、このような複数のバージョンアップデータを混在させて管理できる手法が求められていた。

## 【0008】

【課題を解決するための手段】本発明は、クライアント/サーバ方式のLANシステムにおいて、クライアントが導入するソフトウェアのバージョンアップデータを複数バージョンにわたってサーバに保持し、そのバージョンアップデータをバージョンの順に連結して管理すると共に、クライアントはサーバに保持されたバージョンアップデータを参照して、自クライアントのソフトウェアのバージョン以降のバージョンに順次バージョンアップすることで、上記課題を解決するものである。

## 【0009】

【実施例】図7は、本発明に係るバージョンアップ方法を実施するクライアント/サーバ方式LANシステムの概略構成を示している。71はサーバマシンであり、磁気ディスク等の記憶装置72を持ち、また、CD-ROMドライブ装置等の外付けの第2の記憶装置75を持っていて。記憶装置72には、サーバ固有のデータやプログラムに加えて、さらにクライアントのバージョンアップデータを記憶するバージョンアップデータテーブル73と、クライアント各々の現在のバージョンを管理するクライアント管理テーブル74とを記憶している。また、第2の記憶装置75は外部からバージョンアップデータをサーバ内に入力するために使われる。

【0010】76はクライアントマシンであり、サーバ71と同様に磁気ディスク等の記憶装置77を内蔵している。この記憶装置77にはクライアント上で動作するプログラムやデータ等のソフトウェアが記憶されている。そして、必要に応じてLAN回線70を介してサーバ71のバージョンアップデータを取り込み、クライアントソフトウェアのバージョンアップが行われる。1つまたは複数のクライアント76がLAN回線70に接続可能であり、個々のクライアント76それぞれが、独立にバージョンアップを適宜実施するものである。

【0011】サーバ71に記憶されたクライアント管理テーブル74の例を図5に示す。複数のクライアント76に固有のクライアント番号51に対応付けて、現在それぞれのクライアント76に導入されているクライアントソフトウェアのバージョン情報52を記憶する。このクライアント管理テーブル74はクライアント76から参照可能であり、

クライアント76は、バージョンアップを実施した時、クライアント管理テーブル74の自分のバージョン情報52を更新するものである。

【0012】また、バージョンアップデータテーブル73の例を図4に示す。テーブルの先頭にはリンク情報があり、後続するテーブルのアドレスを指す「次アドレス」と、先行するテーブルのアドレスを指す「前アドレス」が記憶される。これにより各バージョンアップデータがバージョンの順にチェーンされる。そして、リンク情報の後ろにバージョンアップデータの実体が記憶される。但し、先頭のテーブルだけは特別で、リンク情報のみで構成されている。

【0013】同図(a)に示すように、先頭のテーブルの次アドレスは「B」であり、一方前アドレスは、先行するテーブルが存在しないので「終端」コードが記憶されている。後続する2番目のテーブルでは、次アドレスが3番目のテーブルのアドレス「C」であり、前アドレスは「A」である。そして、バージョンアップデータの実体として、バージョン番号等のバージョン情報と、プログラムやデータ等のバージョンアップデータが記憶されている。3番目のテーブルはこの例では末尾のテーブルになり、これは即ち最新のバージョンアップデータである。従って、次アドレスは「終端」である。前アドレスは先行する2番目のテーブルのアドレス「B」である。

【0014】クライアント76は、サーバ71の記憶装置72にあるこのバージョンアップデータテーブル73を参照し、リンク情報の中の次アドレスを追っていくことによって、サーバ71に格納されているすべてのバージョンアップデータのバージョン情報を確認することができ、所望のバージョンをダウンロード可能である。

【0015】次に、実施例の動作を説明する。先ず、バージョンアップデータをサーバ71に格納する場合を図1を参照して説明する。サーバ71は、バージョンアップデータの格納が指示されると、バージョンアップデータテーブル73のうち、先頭のテーブルをカレントテーブル、即ち操作の対象テーブルに設定する(ステップS101)。以下、テーブルの次アドレスや前アドレスと言うときは、カレントテーブルのそれを意味するものとする。そして、そのカレントテーブルの次アドレスが「終端」か否かを判定する(同S102)。終端でなければ、後続するテーブルをカレントテーブルに設定して(同S103)、再びステップS102の判定を実施する。こうして次アドレスが終端になるまでこの処理を繰り返し、最終的に末尾のテーブルをカレントテーブルに設定する。

【0016】次アドレスが終端であると判定されると、カレントテーブルは末尾のバージョンアップデータ、即ち最新のバージョンであるということである。そこでサーバ71は、新バージョンのバージョンアップデータを第2の記憶装置75から内部の記憶装置72にコピーしておい

て(同S104)、カレントテーブルの次アドレスにはコピーした新バージョンのアドレスを記憶し(同S105)、新バージョンの前アドレスにはカレントテーブルのアドレスを記憶し(同S106)、そして、新バージョンの次アドレスには終端コードを記憶する(同S107)。このようにして、コピーした新バージョンのバージョンアップデータを、バージョンアップデータテーブル73の末尾に連結するのである。

【0017】図4(b)は、同図(a)の状態に新バージョンが追加された様子を示している。同図(a)において末尾にあったテーブル、即ちカレントテーブルの次アドレスには最新バージョンのアドレス「D」が記憶され、最新バージョンの前アドレスにはアドレス「C」、同次アドレスには「終端」が記憶されている。

【0018】次に、クライアント76がバージョンアップを実施する場合を図2を参照して説明する。クライアント76は、サーバ71のバージョンアップデータテーブル73を参照し、先頭テーブルをカレントテーブルに設定して(ステップS201)、その次アドレスが終端かどうか判定する(同S202)。終端でなければ、後続のテーブルをカレントテーブルに設定しておいて(同S203)、カレントテーブルのバージョン情報と自身のクライアントソフトウェアのバージョンとを比較し(同S204)、バージョンアップが必要か否かを判定する。具体的には、カレントテーブルのバージョンがクライアント自身のそれより新しいか否かを判定して、新しければバージョンアップが必要と判定する。バージョンアップが必要な場合は、クライアント76はサーバ71から該当するバージョンアップデータ、即ちカレントテーブルに記憶されているバージョンアップデータの実体データをダウンロードし、バージョンアップを実施する(同S205)。そして今度は、サーバ71のクライアント管理テーブル74を参照し、自身のクライアント番号に対応するバージョン情報52を、バージョンアップしたバージョンに更新する(同S206)。つまり、クライアント管理テーブル74の該当箇所に、カレントテーブルのバージョン情報を記録するのである。その後、ステップS202に復帰する。

【0019】一方、ステップS204において、カレントテーブルのバージョンがクライアント自身のそれより古い場合は、バージョンアップが必要と判定してステップS202に直接復帰する。そして、次アドレス=終端となって後続するテーブルが無くなったら処理を終了する。

【0020】このように、サーバ71の記憶装置72上に連結して格納されているバージョンアップデータのバージョンを先頭から順に参照し、クライアント自身のバージョンより新しいものが見つかる度にバージョンアップを繰り返すのである。

【0021】バージョンアップデータはバージョンの古いものから順に新しいものが後ろに連結され、末尾に最

新のバージョンアップデータがある。クライアント76は、サーバ1にある最も古いバージョンのバージョンアップデータから開始して、1つ1つ自身のバージョンと比較してそれらが古い間は無視し、自身のバージョンより新しいものが現れてからは、最新のバージョンに到るまで全世代のバージョンにわたって、それらバージョンアップデータを使って順にバージョンアップを実施するのである。その結果、2世代以上遅れてバージョンアップする場合にも、最新バージョンに含まれない旧世代の差分データをも漏れなく取込んでバージョンアップできるものである。

【0022】次に、不要になったバージョンアップデータを削除する場合を図3を参照して説明する。サーバ1は先ず、バージョンアップデータテーブル73の先頭テーブルをカレントテーブルに設定する(ステップS301)。そして、次アドレスが終端か否か判定する(同S302)。終端でなければ、後続するテーブルをカレントテーブルに設定し、その次アドレスと前アドレスを別途記憶しておく(同S303)。次にカレントテーブルのバージョン情報と、クライアント管理テーブル74に記憶されている全クライアントのバージョン情報52とを比較し(同S304)、いずれのクライアント76もカレントテーブルのバージョンに既にバージョンアップ済みであるか判定する。ここでは、クライアント76のバージョンがカレントテーブルのそれより新しいか同じである場合は、カレントテーブルのバージョンに既に更新済みであると判定するのである。

【0023】そこで、更新済みの場合には、カレントテーブルであるバージョンアップデータを削除し(同S305)、別途記憶しておいたカレントテーブルの次アドレスを先行するテーブルの次アドレスに記憶する(同S306)。そして、その次アドレスが終端か否か判定し(同S307)、終端でなければ、記憶してあるカレントテーブルの前アドレスを後続するテーブルの前アドレスに記憶する(同S305)。その後、ステップS302に復帰する。尚、末尾のテーブル即ち最新のバージョンアップデータを削除したときは、後続のテーブルは存在しないので、ステップS307から同S302に復帰するようしている。

【0024】他方、ステップS304において、1つでも未更新のクライアント76があると判定すると、直接ステップS302に復帰する。こうして次アドレスが終端になるまでこの処理を繰り返し、バージョンの古いものから順に、全クライアントでバージョンアップが済んでいるバージョンアップデータを、サーバ1から削除していくのである。そして、次アドレス=終端となったところで処理を終了する。

【0025】尚、ステップS304で未更新ありと判定したとき、ステップS302に復帰せずに、その時点での処理を終了してもよい。バージョンアップデータはバ

ジョンの古い順に連結されていて、しかも、クライアント76は途中のバージョンを飛ばしてバージョンアップをすることはない。従って、クライアントが未導入していないバージョンがあれば、それ以降のバージョンも同じくクライアントが未導入であると判定して差し支えないからである。そして、さらに、この古いバージョンの削除処理は、先に述べた新しいバージョンのサーバ1への格納処理と同期する必要はない。また、クライアント76のバージョンアップ処理とも同期の必要はなく、サーバ1の都合のよい時に任意に実施できるものである。

【0026】ところで、本実施例では、図4に示したように、バージョンアップデータをテーブル先頭に付加したリンク情報で連結して管理しているが、この方法に限るものではない。図6にバージョンアップデータテーブルの別の実施例を示す。バージョンアップデータテーブル60は、古いものから順にバージョン番号62と、そのバージョンのバージョンアップデータの実体が記憶されているアドレス63とを対にして記憶する。テーブルのレコード番号61から分かるように、この例では現在サーバ1にn個のバージョンアップデータが保持されている。

尚、このレコード番号61は、図5に示すクライアント管理テーブル74のクライアント番号51とは何の関係も無い。

【0027】そして、クライアント76は、サーバ1にあるこのバージョンアップデータテーブル60を参照し、先頭レコードのバージョン番号62と自身のクライアントソフトウェアのバージョンとを比較して、バージョンアップが必要か否かを判定する。具体的には、レコードに記憶されたバージョン62がクライアント自身のそれより新しいか否かを判定して、新しければバージョンアップが必要と判定する。バージョンアップ要の場合は、対応するアドレス63を参照してバージョンアップデータの実体データをダウンロードし、バージョンアップを実施する。そして次に、サーバ1のクライアント管理テーブル74を参照し、自身のクライアント番号51に対応するバージョン情報52を、バージョンアップ後のバージョンに更新する。つまり、クライアント管理テーブル74の該当箇所に、導入したバージョン番号を記録するのである。クライアント76はこの処理をすべてのレコードにわたって順に実施し、クライアント自身のバージョンより新しいものが見つかる度に、バージョンアップを繰り返せばよい。

【0028】一方、不要になった古いバージョンアップデータを削除するときサーバは、バージョンアップデータテーブル60の先頭レコードから順に、そこに記録されているバージョン番号62と、クライアント管理テーブル74に記憶されている全クライアントのバージョン情報52とを比較して、どのクライアント76もそのバージョンに既にバージョンアップ済みであるか否か判定する。そして、バージョンアップ済みであるときは、当該レコード

のアドレス63を参照して、そのバージョン番号62に対応するバージョンアップデータの実体を削除すると共に、当該レコードも削除する。その後、後続のレコードのレコード番号61を付け替えておいて、以降のレコードに対しても同様の処理を繰り返せばよい。

【0029】ところで、クライアントにおけるバージョンアップ処理は、サーバとの関係においても、またクライアント相互の関係においても全く独立に実施できるものである。即ち、サーバにおける新バージョンの格納処理、及び古いバージョンの削除処理と、クライアントのバージョンアップ処理とは同期をとる必要はなく、それぞれ任意に実施してよい。また、クライアント相互においても同期を取ってバージョンアップ処理を行う必要はない。個々のクライアントが任意に実施すればよいものである。

#### 【0030】

【発明の効果】本発明によれば、複数のバージョンのクライアントソフトをそのバージョンの順に連結してサーバに保持することができる。その為、クライアントはそれぞれに必要なバージョンを容易に探すことができ、所望のバージョンデータを任意にダウンロードしてバージョンアップができる。そして、クライアントは相互に独立にバージョンアップができるので、LANで結ばれたクライアント/サーバシステムの内部に、クライアントソフトの複数バージョンの混在を許容する事ができる。また、サーバは古いバージョンのクライアントソフトのバージョンアップデータを、クライアントのバージョン\*

\*アップ処理とは同期に削除できるようになる。さらに、バージョンアップデータが差分データで構成されている場合、クライアントはバージョンの古い順に1つ1つバージョンアップ処理するので、漏れなく正しく最新バージョンに移行できる。

#### 【図面の簡単な説明】

【図1】実施例におけるサーバへの新バージョンの格納手順を示す図である。

10 【図2】実施例におけるクライアントのバージョンアップ処理の手順を示す図である。

【図3】実施例におけるサーバから旧バージョンを削除する手順を示す図である。

【図4】実施例のバージョンアップデータテーブルを示す図である。

【図5】実施例のクライアント管理テーブルを示す図である。

【図6】バージョンアップデータテーブルの別の実施例を示す図である。

20 【図7】実施例のクライアント/サーバシステムの構成を示すブロック図である。

#### 【符号の説明】

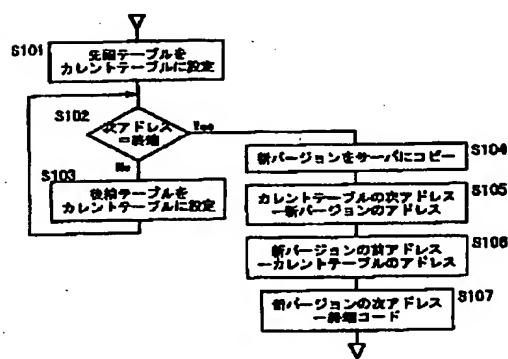
70 LAN回線 71 サーバ 72 記憶装置

73 バージョンアップデータテーブル

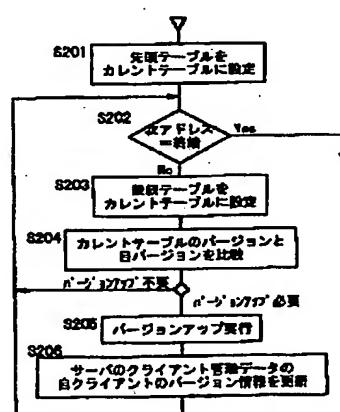
74 クライアント管理テーブル

76 クライアント 77 記憶装置

【図1】



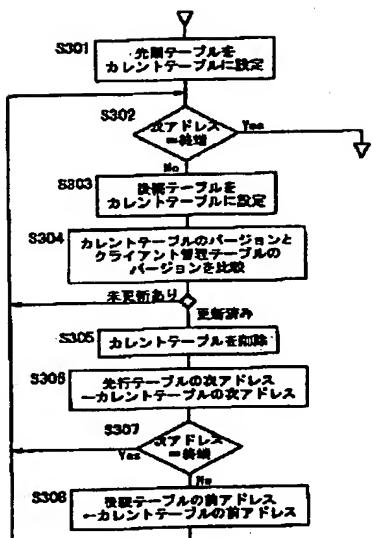
【図2】



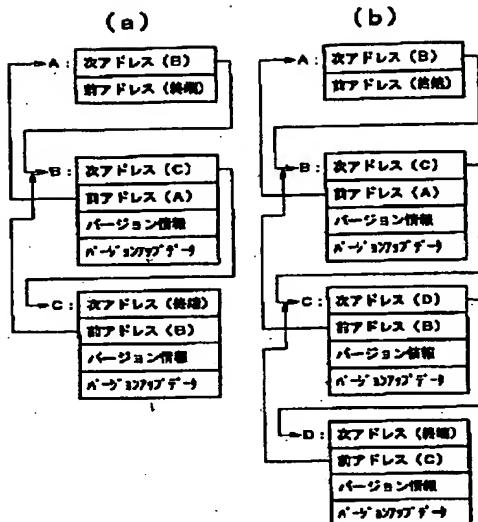
【図5】

51 クライアント番号	52 バージョン情報
1	1. 1. 0
2	1. 0. 0
⋮	⋮
N	1. 5. 0

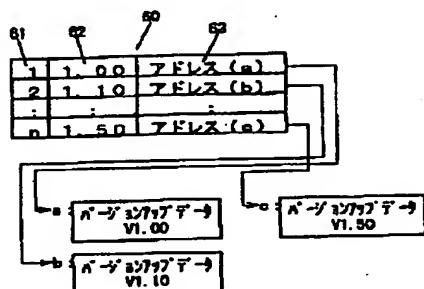
【図3】



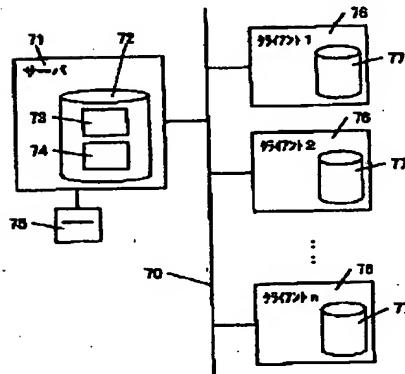
[図4]



〔図6〕



[图7]



Handwritten notes and diagrams for V1, I1, and I2.

Top left:  $V_1, I_1, 0$

Middle left:  $I_1, I_2$  (inside a rectangular box)

Bottom left:  $I_1, 2$

Top right:  $I$  (with a downward arrow)

Bottom right:  $I$  (with a downward arrow)

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**